CAN 기반 로봇 제어

목차

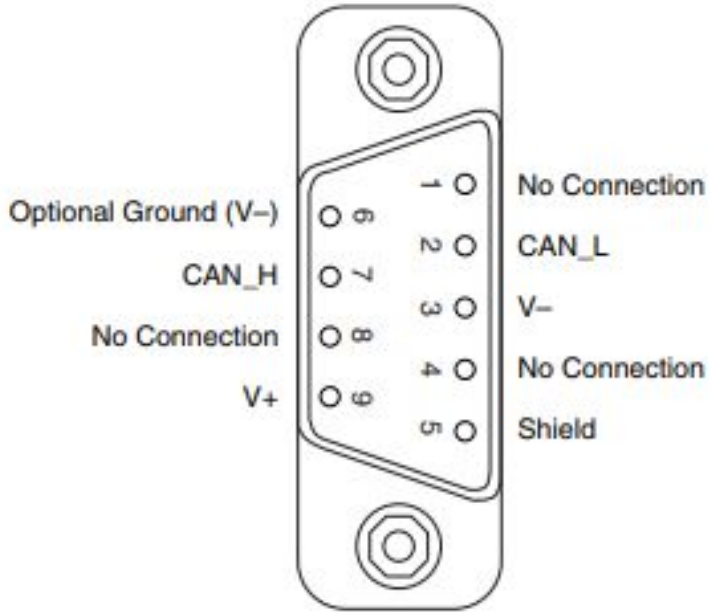
01

About CAN

ROS

WeGo

- **CAN**
- CAN 통신은 Controller Area Network의 줄임말이며, 호스트 없이, MCU 및 기타 장치들이 서로 통신을 하기 위해 설계된 표준 통신 규격으로, 일반적으로 차량에서 많이 사용되었다.
- 1983년 Bosch사에서 최초 개발되었으며, 1986년에 소개된 이후로, 대부분의 자동차에서 사용하고 있다.



01. CAN

- **CAN**
- CAN Port로 연결하여 통신을 할 수 있으며, 내부 네트워크에 흐르는 데이터를 주소를 통해서 구분하게 된다.
- 각 데이터 주소에 어떤 의미의 값이 전달이 되는지를 알아야하며, 이를 저장해놓은 파일이 CAN Database or .DBC 파일이다.
- 일반적인 차량 내부의 DBC파일은 국가 재산으로 일반적으로는 공개되지 않고있다.
- 데이터를 송신 및 수신할 때도, DBC파일을 이용하여, 어떠한 주 송신할지 및 어떠한 주소의 데이터를 확인할 지 볼 수 있다.
- PC에서 로깅을 위해서 CAN to USB 장치를 사용한다.

ROS

WeGo

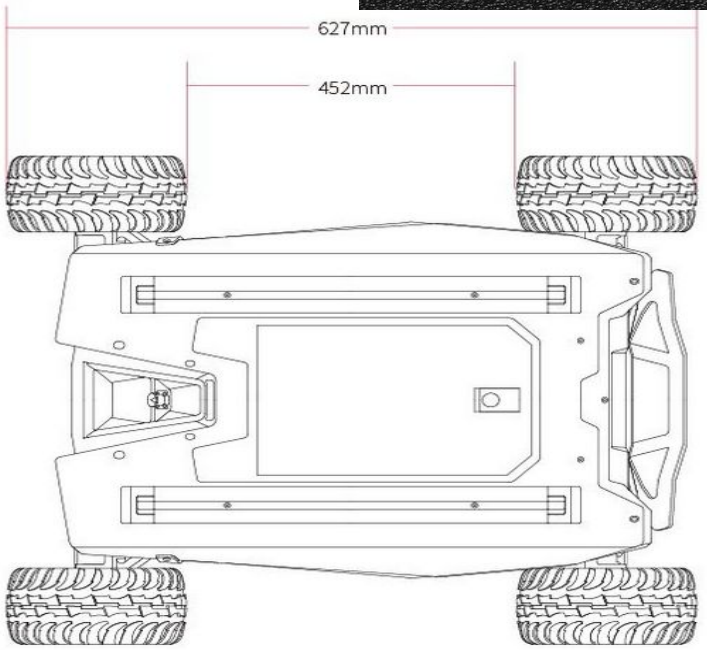
02

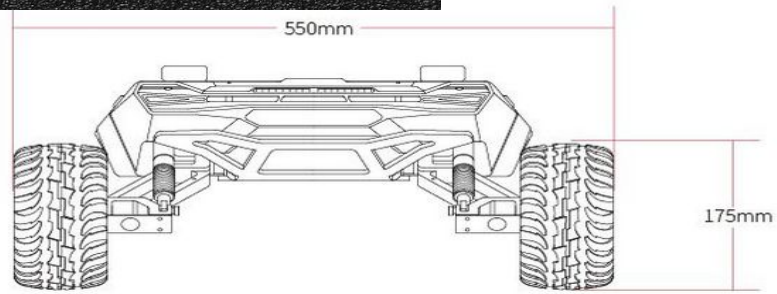
Scout mini

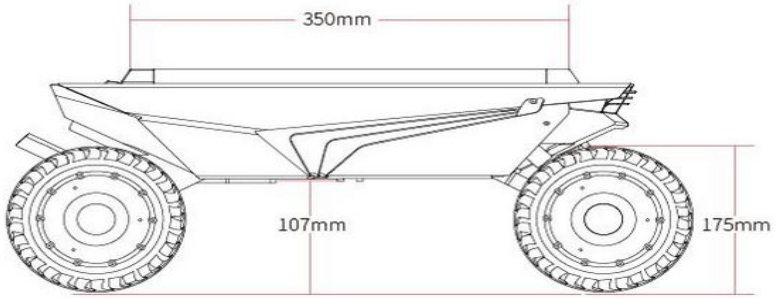
ROS

WeGo

- Scout mini







ROS

WeGo

02. Scout mini

- Scout mini

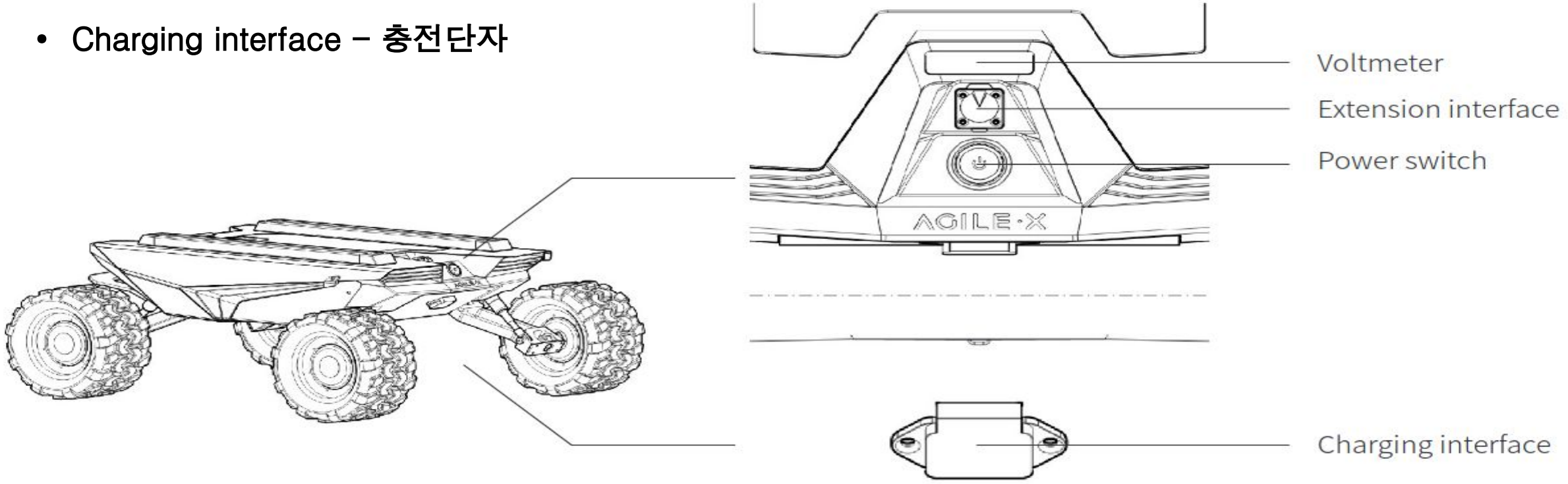
크기	627 X 550 X 252 mm
운행 형태	4륜 구동, Differential-Drive Model
축거(Wheelbase)	452mm
배터리 동작 온도	-20~60℃
충전 시간	2시간
최저지상고	107mm
최소 회전 반경	0m
배터리	24V / 15Ah
최대 속도	10km/h
통신 환경	Standard CAN / RS232

ROS

WeGo

02. Scout mini

- Scout mini
- Voltmeter – 현재 배터리 상태 및 잔량 확인
- Extension interface – CAN or 24V 전원 사용 가능
- Power switch – Scout mini 전원
- Charging interface – 충전단자



ROS

WeGo

02. Scout mini

- Scout mini Controller
- 전원 On / Off를 위해 7, 8 버튼을 길게 입력(1, 2, 3, 4는 위 올린 상태로 구동)
- 수동 모드 조작을 위해 2를 중앙으로 이동한 후, 5를 이용하여 전, 후방 이동, 6을 이용하여 좌우 회전 조작 가능
- Serial or CAN을 이용한 주행을 위해서는 2를 위(CAN) 또는 아래(Serial)로 이동하면 자율 주행 모드 실행 가능
- 3을 이용하여, 수동 모드 시, Scout mini의 조명을 변경 가능
- 4를 이용하여, Scout mini의 최대 이동 속도 변경 가능(Speed Mode, Normal Mode)

1. Lever SWA
2. Lever SWB
3. Lever SWC
4. Lever SWD
5. Left rocker
6. Right rocker
7. Power switch key 1
8. Power switch key 2
9. Mobile/Tablet fixing support interface
10. Ring interface
11. LCD panel

*When the user gets the RC transmitter, the settings have been available without having to be set separately.

03

CAN 기반 제어

ROS

WeGo

Control Scout mini Using ROS

- Install ROS Package & Build Package
- $ mkdir -p ~/catkin_ws/src
- $ cd ~/catkin_ws/src
- $ git clone https://github.com/agilexrobotics/scout_ros.git
- Install Dependencies
- $ sudo apt install ros-melodic-teleop-twist-keyboard
- $ sudo apt install ros-melodic-joint-state-publisher-gui
- $ sudo apt install ros-melodic-ros-controllers
- $ cd ~/catkin_ws
- $ catkin_make

- Setup UART port
- $ sudo usermod -a -G dialout $USER
- $ sudo chmod 666 /dev/tty*
- Setup CAN-To-USB
- $ sudo modprobe gs_usb
- $ sudo ip link set can0 up type can bitrate 500000
- $ ifconfig -a (설정 확인을 위한 부분)
- $ sudo apt install can-utils (최초 실행에만 필요)
- $ candump can0 (데이터 입출력 확인을 위해 사용)
- $ rosrun scout_bringup setup_can2usb.bash (위의 내용을 한 번에 실행)
- $ rosrun scout_bringup bringup_can2usb.bash (재부팅 or USB 재연결시 실행)

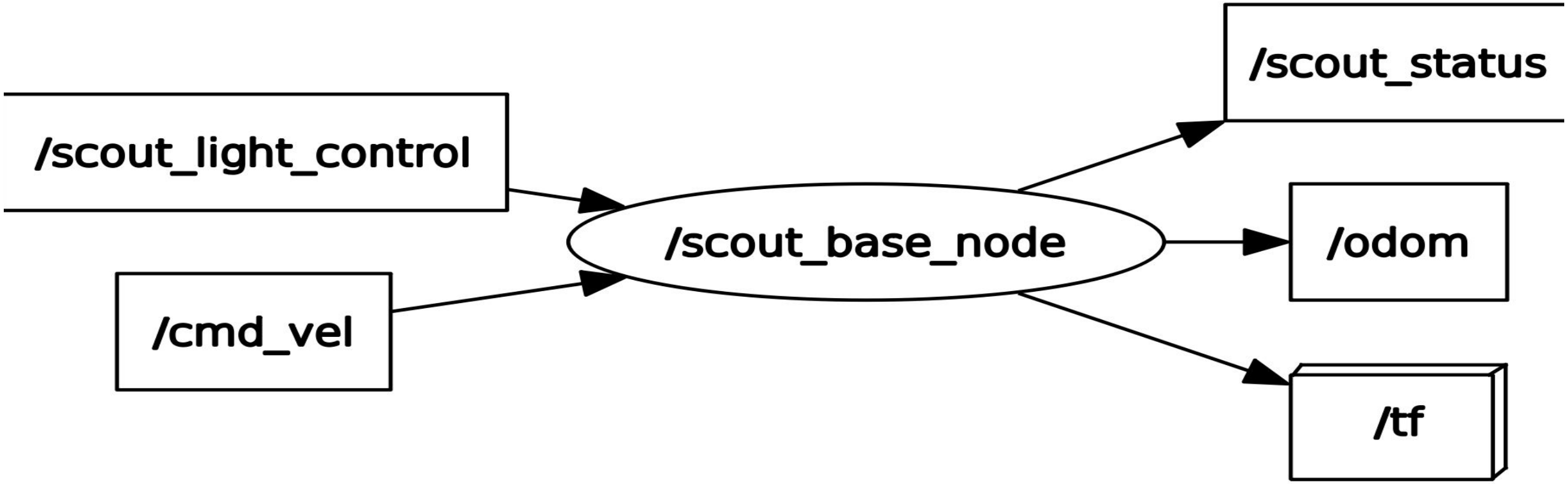
Control Scout mini Using ROS

- Start Node
- $ roslaunch scout_bringup scout_minimal.launch (using CAN)
- $ roslaunch scout_bringup scout_minimal_uart.launch (using RS232)

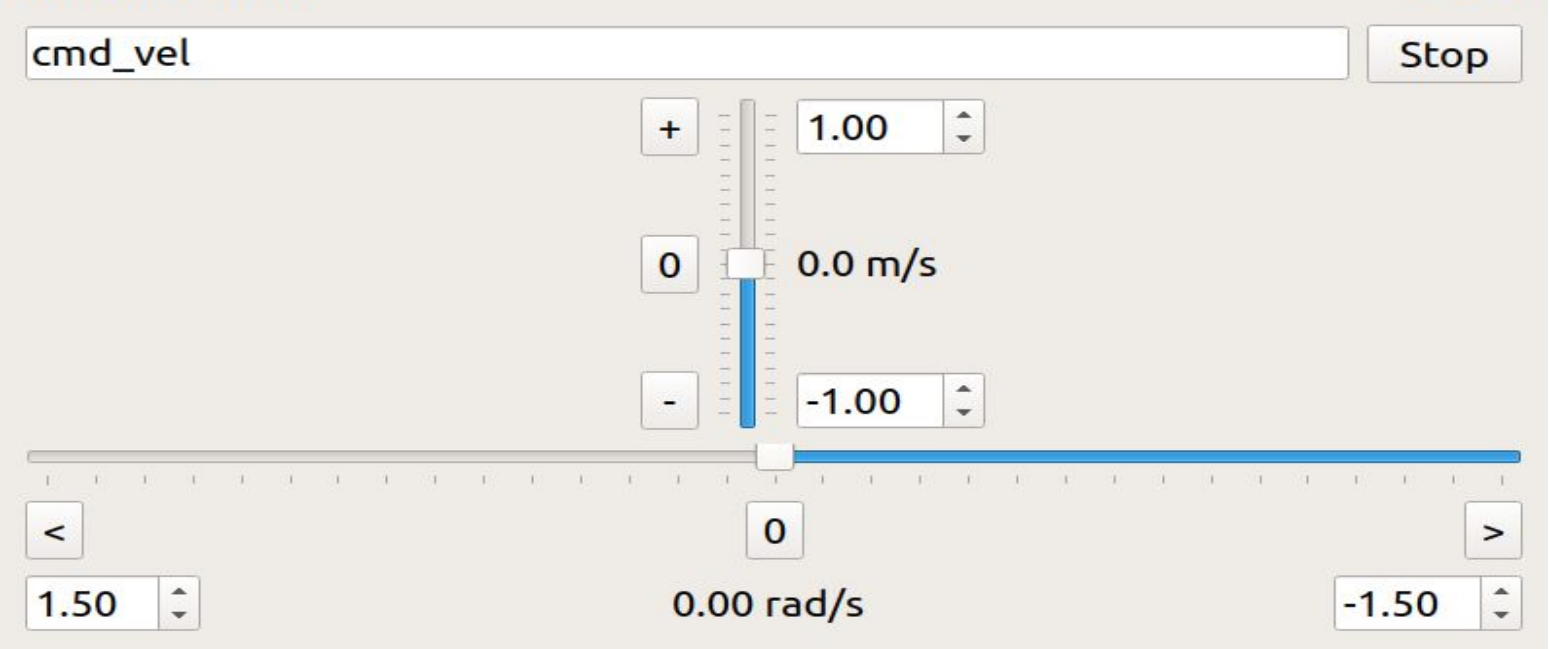
- Start Node

```
started roslaunch server http://wego-GF63-Thin-10SCXR:37973/

SUMMARY
========

PARAMETERS
 * /rosdistro: melodic
 * /rosversion: 1.14.10
 * /scout_base_node/base_frame: base_link
 * /scout_base_node/odom_frame: odom
 * /scout_base_node/port_name: /dev/ttyUSB0
 * /scout_base_node/simulated_robot: False

NODES
  /
    scout_base_node (scout_base/scout_base_node)

auto-starting new master
process[master]: started with pid [5765]
ROS_MASTER_URI=http://localhost:11311

setting /run_id to d2eb247e-353b-11eb-8ba4-d83bbf196793
process[rosout-1]: started with pid [5791]
started core service [/rosout]
process[scout_base_node-2]: started with pid [5797]
connection: 0 , device: /dev/ttyUSB0 @ 115200bps
[ INFO] [1606981663.367117586]: Using UART to talk with the robot
```

- 자동 모드 테스트
- 테스트 전 rqt graph 실행하여, driver가 cmd_vel을 subscribe하고 있는지 확인
- $ rqt_graph

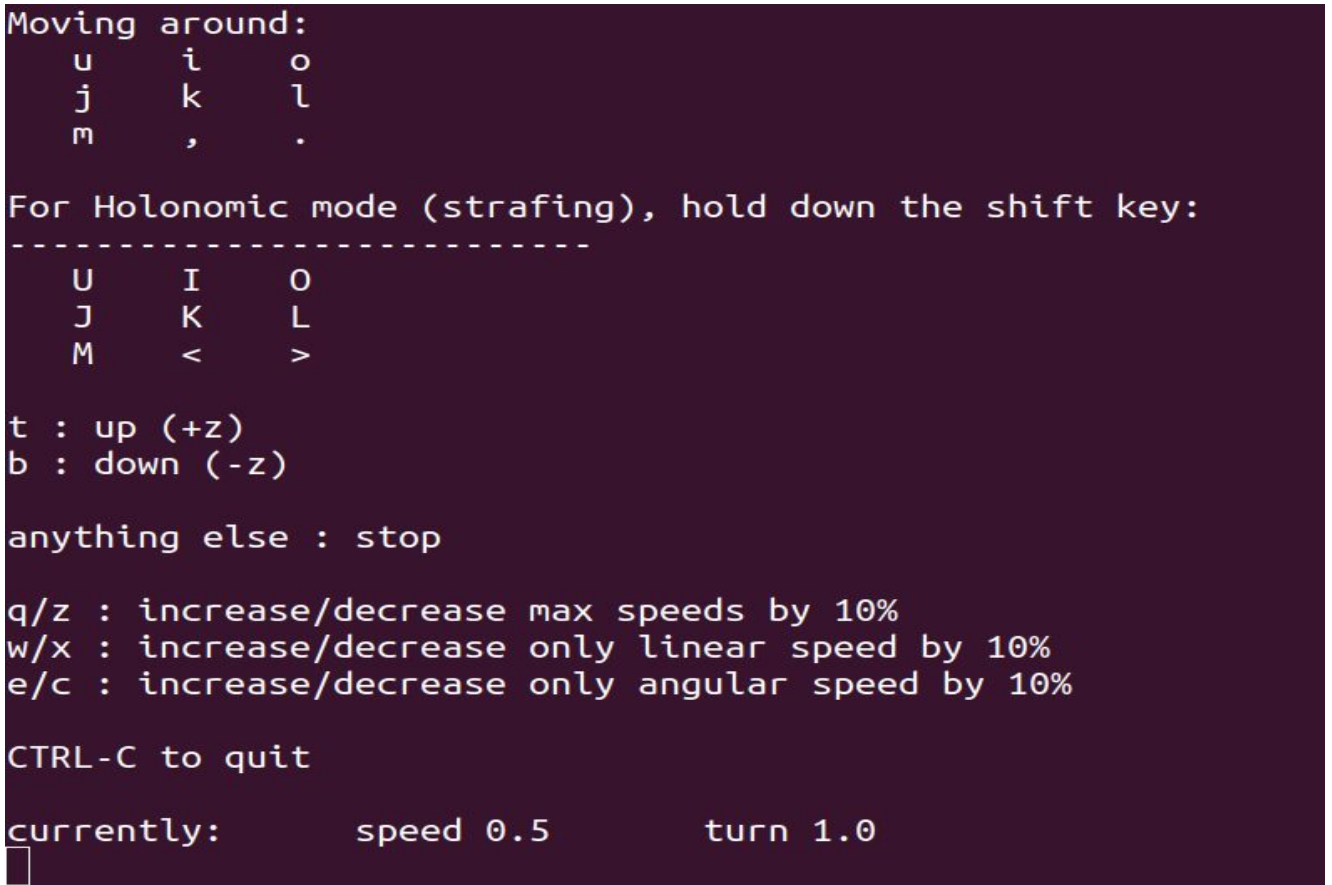


- 자동 모드 테스트
- 방법 1 : Robot-Steering 이용
- $ sudo apt install ros-melodic-rqt-robot-steering
- $ rosrun rqt_robot_steering rqt_robot_steering
- 위의 cmd_vel을 확인 후, 조종기의 모드를 변경한 후, 아래의 값 변경을 통해 제어 가능



03 Control Scout mini Using ROS

- 자동 모드 테스트
- 방법 2 : Keyboard Teleop 이용
- $ roslaunch scout_bringup scout_teleop_keyboard.launch
- 아래의 그림에서 설명을 통해 제어 가능(q, z, w, x, e, c를 통해 속도 변경)
- u, i, o, j, k, l, m, , , .을 통해 이동 방향 및 회전 제어 가능



ROS

WeGo

04

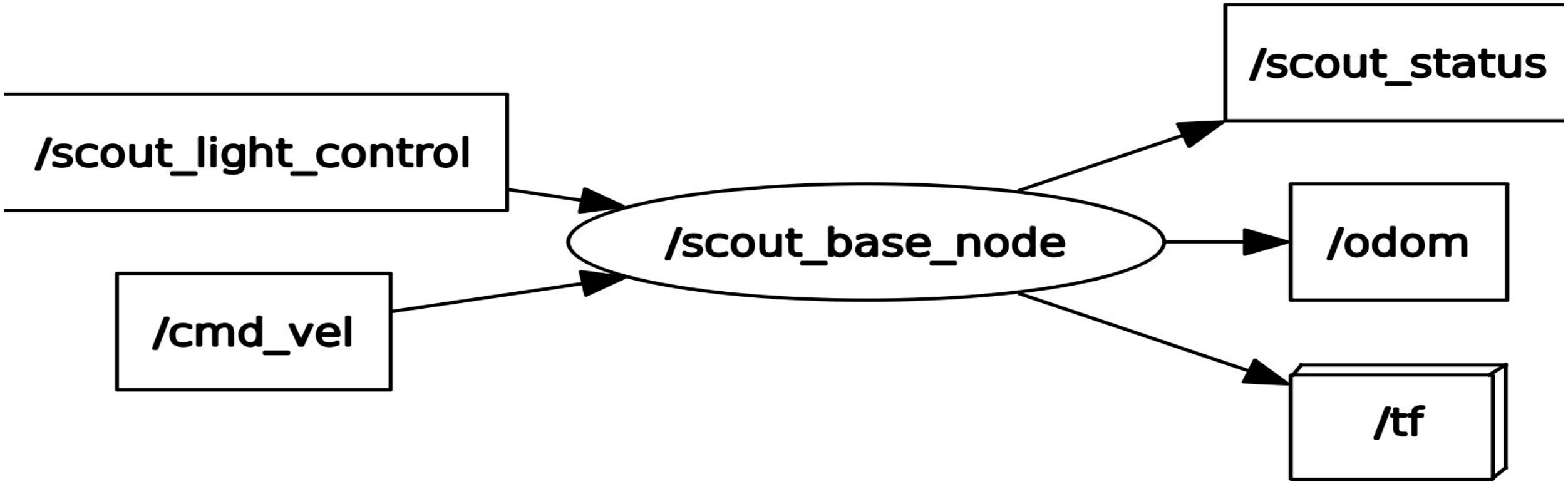
Scout mini ROS Topic

ROS

WeGo

04 Scout mini ROS Topic

- Scout mini base Node
- Publishing Topic : /scout_status, /odom, /tf
- Subscribing Topic : /scout_light_control, /cmd_vel



04 Scout mini ROS Topic

- /scout_status Msg Type (FR = Front Right, RL = Rear Left)
- std_msgs/Header header
    - uint32 seq
    - time stamp
    - string frame_id
  - float64 linear_velocity
  - float64 angular_velocity
  - uint8 base_state
  - uint8 control_mode
  - uint16 fault_code
  - float64 battery_voltage
  - scout_msgs/ScoutMotorState[4] motor_states (FR = 0, FL = 1, RR = 2, RL = 3)
    - float64 current
    - float64 rpm
    - float64 temperature
  - bool light_control_enabled
  - scout_msgs/ScoutLightState front_light_state
    - uint8 mode
    - uint8 custom_value
  - scout_msgs/ScoutLightState rear_light_state
    - uint8 mode
    - uint8 custom_value

04 Scout mini ROS Topic

- /scout_status → 실제 Scout mini의 상태를 출력
- linear_velocity → Scout mini의 속도
- angular_velocity → Scout mini의 회전 속도
- base_state → 사용하지 않음
- control_mode : 0이면 수동, 1이면 CAN, 2이면 RS232
- fault_code : 0이면 정상, 문제 발생 시 다른 숫자
- motor_states : 각 모터의 상태 출력
- light_control_enabled : 자동 모드 시, light 제어 상태 확인 가능
- front_light_state – mode : 0이면 꺼짐, 1이면 고정값, 2이면 숨쉬기 모드
- rear_light_state : 후방 light 없음

04 Scout mini ROS Topic

- /odom Msg Type
- std_msgs/Header header
    - uint32 seq
    - time stamp
    - string frame_id
  - child_frame_id
  - pose
    - pose
      - position
        - x,y,z
      - orientation
        - x,y,z,w
      - covariance
  - twist
    - twist
      - linear
        - x,y,z
      - angular
        - x,y,z
    - covariance

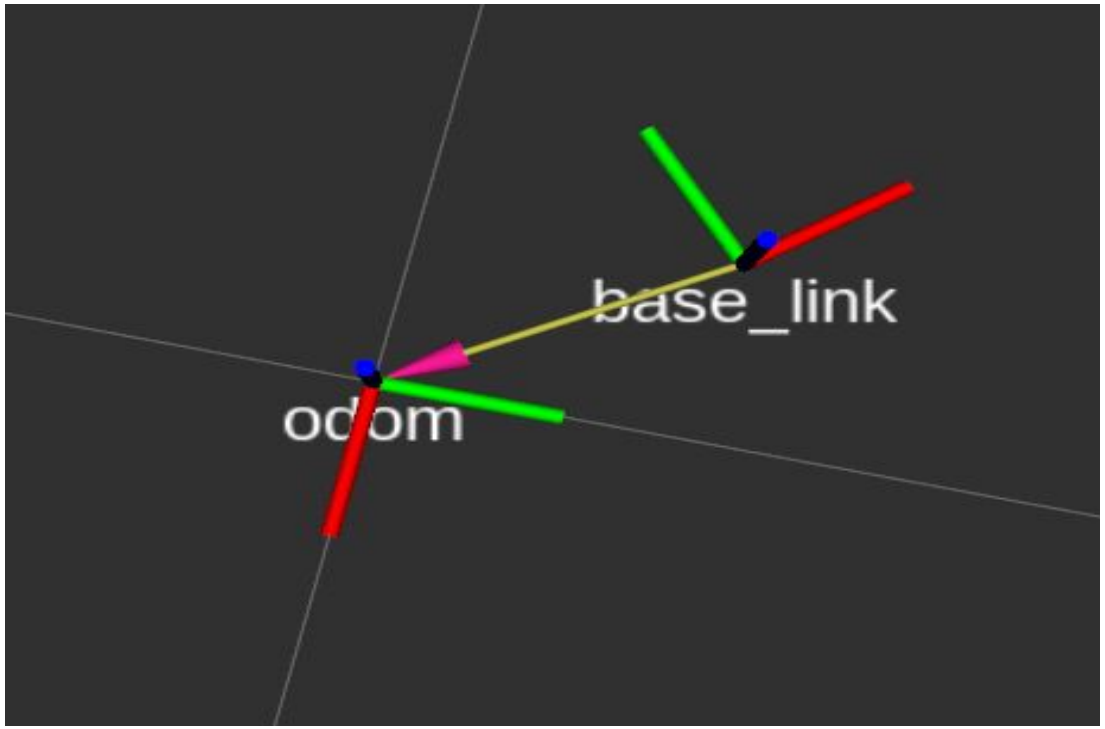
04 Scout mini ROS Topic

- /odom → 실제 Scout mini가 ROS와 연결된 후, 바퀴를 통해 이동한 현재 위치 및 현재 이동 상태를 확인 가능
- pose : 출발 이후, 변화한 x, y 및 yaw에 해당하는 값 확인 가능
- pose – covariance : 사용하지 않음
- twist – 현재 scout mini의 선속도 및 회전 속도 확인 가능
- twist – covariance : 사용하지 않음
- rviz를 통해 출발 위치인 odom과 그 child frame인 base_link를 통해, 출발 지점에서 변화한 정도를 확인 가능

04 Scout mini ROS Topic

- /tf Msg Type
- std_msgs/Header header
    - uint32 seq
    - time stamp
    - string frame_id
  - child_frame_id
  - transform
    - translation
      - x, y, z
    - rotation
      - x, y, z, w

- /tf → ROS Frame 사이의 변화량에 대한 값 확인 가능
- child_frame_id : 변환할 frame의 이름
- transform : 부모 Frame에서 자식 Frame 사이의 변화량 확인 가능
- transform – translation : 부모 자식 Framex, y, z 사이의 변화량 확인 가능
- transform – rotation : 부모 자식 Frame 사이의 roll, pitch, yaw 변화량 확인 가능
- rviz의 tf 및 axes를 통해서 변화된 값을 직접 시각적으로 확인 가능



04 Scout mini ROS Topic

- /scout_light_control Msg Type
- bool enable_cmd_light_control
- uint8 front_mode
- uint8 front_custom_value
- uint8 rear_mode
- uint8 rear_custom_value

04 Scout mini ROS Topic

- /scout_light_control → 자동 모드에서 scout의 전방 light control 가능
- enable_cmd_light_control : 제어할지 안할지를 입력 (true, false)
- front_mode : 0일 경우 Off, 1일 경우, 현재 값으로 고정, 2일 경우, 숨쉬기 모드, 3일 경우 아래에서 입력하는 custom value로 설정
- front_custom_value : Mode가 3일 경우 사용하는 Value
- rear_mode – 사용할 수 없음

04 Scout mini ROS Topic

- /cmd_vel Msg Type
- geometry_msgs/Vector3 linear
  - float64 x
  - float64 y
  - float64 z
  -
- geometry_msgs/Vector3 angular
  - float64 x
  - float64 y
  - float64 z
  -

04 Scout mini ROS Topic

- /cmd_vel → 자동 모드 시, 속도 및 회전 속도 제어
- linear – x값만 사용하며, m/s단위로 이동 속도 입력
- angular – z값만 사용하며, 회전 속도를 rad/s단위로 입력

WeGo

Q & A

go.support@wego-robotics.com

go.sales@wego-robotics.com